OmDayal Group of Institutions
ENGINEERING | ARCHITECTURE | MANAGEMENT

Name: Soumyodip Thanadar
Roll no: 27500122024
Name: Arpan Maity
Roll no: 27500122014
Name: Prasanta Adak
Roll no: 27500123059
Dept. : Computer Science And Engineering
Name of project: Library Management System
Year:3rd year

What is python Programming?

Python is a high-level, interpreted programming language known for its simplicity and readability. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often used for web development, data analysis, artificial intelligence (AI), machine learning (ML), automation, and more.

Real-life Applications of Python:

Web Development

Data Science & Analytics:

Artificial Intelligence (AI) &

Machine Learning (ML)

Automation

Why SQL for Our Library Management System?

We are using SQL with Python for our Library Management System because:

•**Structured Data**: SQL handles structured data with clear relationships, perfect for books, members, and transactions.

•**Ease of Use:** SQL's syntax is simple and integrates seamlessly with Python libraries like SQLite3.

•**Reliability**: SQL databases are ACID-compliant, ensuring data accuracy and consistency.

•**Scalability**: SQL can handle growing data as the library expands.

History of SQL:

•1970s: The concept of a relational database system was introduced by Dr. Edgar F. Codd, a computer scientist working at IBM. He proposed the relational model for managing data in a structured format.

•1974: IBM developed SEQUEL (Structured English Query Language), a language designed to manage and query relational databases, based on Dr. Codd's theories.

•1979: IBM's System R was the first relational database management system (RDBMS) to implement SQL.

•1986: The American National Standards Institute (ANSI) standardized SQL, which led to its widespread adoption. This version was called SQL-86.

•1989-1992: Several versions of SQL were released, including SQL-89 and SQL-92, with improvements in functionality and compliance.

•1999: The SQL:1999 standard added object-oriented features and greater support for procedural programming.

•2000s and Beyond: SQL continued to evolve, with modern versions like SQL:2003, SQL:2008, and newer additions, introducing features for XML handling, window functions, and mor

SQLite 3 is a widely used, lightweight, and self-contained relational database management system (RDBMS). It is unique in that it is serverless, zero-configuration, and fully embedded into the application using it. SQLite uses a simple file-based architecture, storing the entire database (including tables, indexes, and the data) in a single disk file.

Key Features of SQLite 3:

Self-Contained:

Serverless:

Transactional:

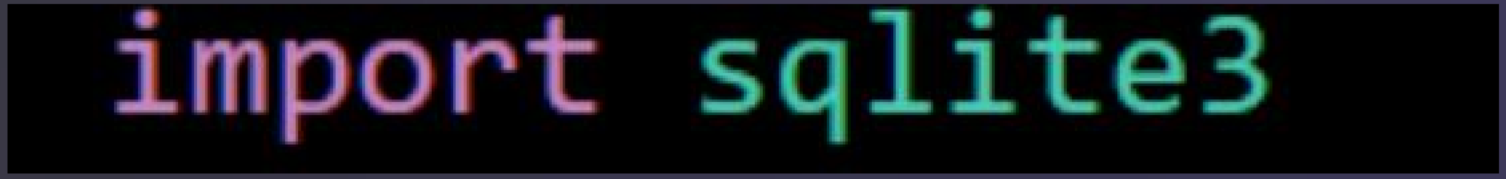
Cross-Platform:

Open Source

- Embedded databases in mobile and desktop applications.
- Prototyping and testing for larger database systems.

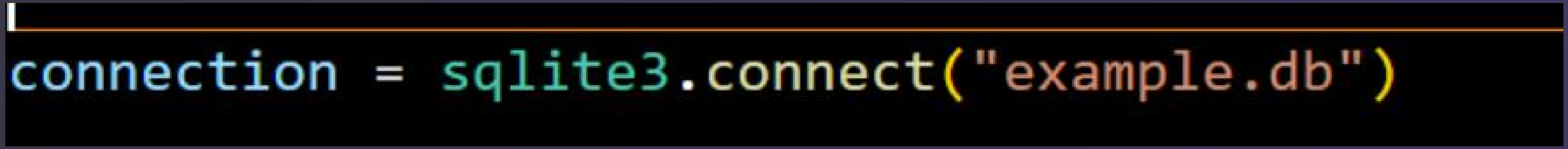
Use the connect method to connect to an SQLite database. If the specified database file doesn't exist, it will be created

Python's sqlite3 module is used to interact with SQLite databases.



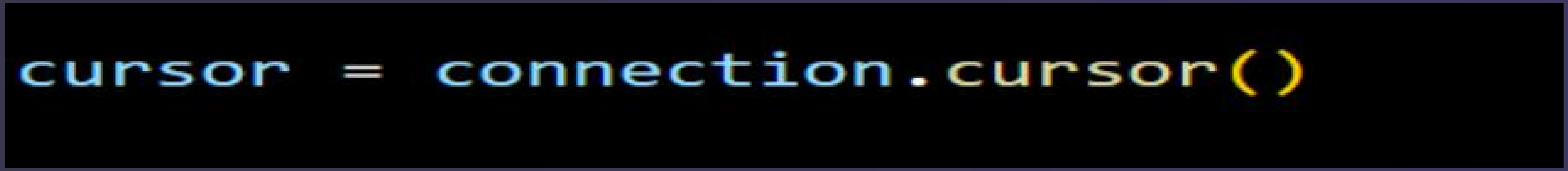
```
import sqlite3
```

Use the connect method to connect to an Sqlite database



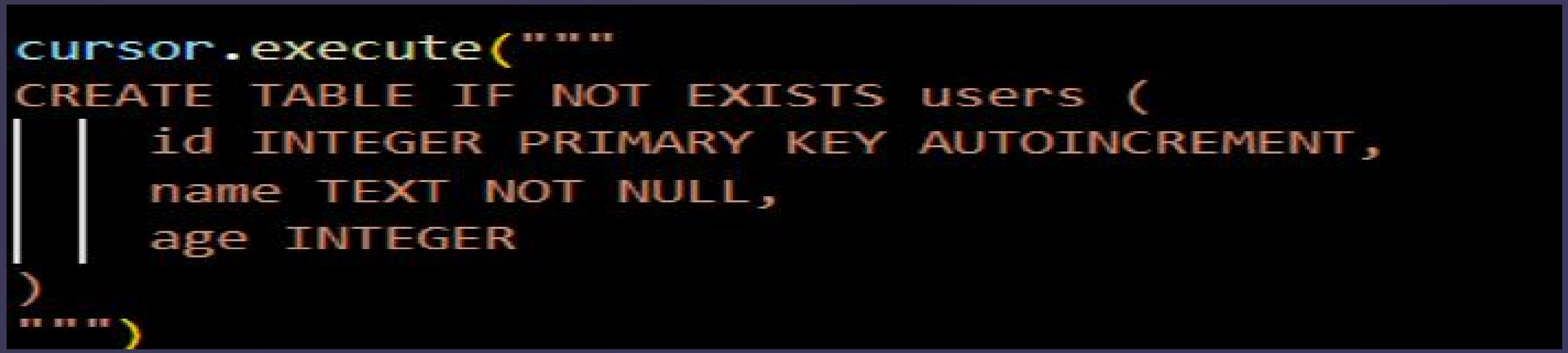
```
connection = sqlite3.connect("example.db")
```

A cursor allows you to execute SQL commands and fetch data.

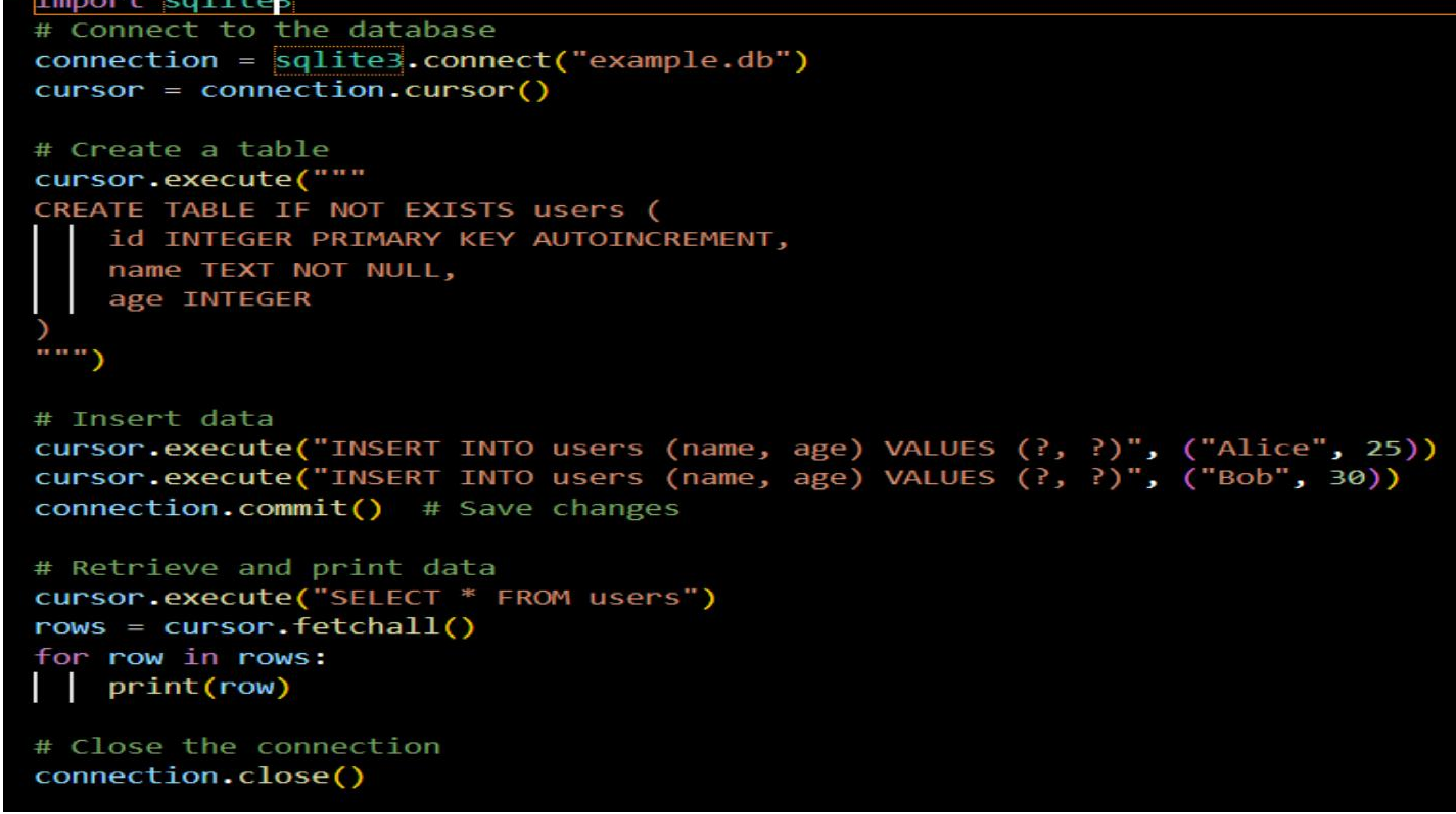


```
cursor = connection.cursor()
```

Creating table



```
cursor.execute("""
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    age INTEGER
)
""")
```



```python
import sqlite3
# Connect to the database
connection = sqlite3.connect("example.db")
cursor = connection.cursor()

# Create a table
cursor.execute("""
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    age INTEGER
)
""")

# Insert data
cursor.execute("INSERT INTO users (name, age) VALUES (?, ?)", ("Alice", 25))
cursor.execute("INSERT INTO users (name, age) VALUES (?, ?)", ("Bob", 30))
connection.commit()  # Save changes

# Retrieve and print data
cursor.execute("SELECT * FROM users")
rows = cursor.fetchall()
for row in rows:
    print(row)

# Close the connection
connection.close()
```

e

PyQt is a set of Python bindings for the Qt application framework. It is widely used to develop desktop applications with graphical user interfaces (GUIs). PyQt combines the power of Python with the robust capabilities of Qt, enabling developers to create cross-platform applications with ease.

Features of PyQt

Cross-Platform Support: Works on Windows, macOS, and Linux.

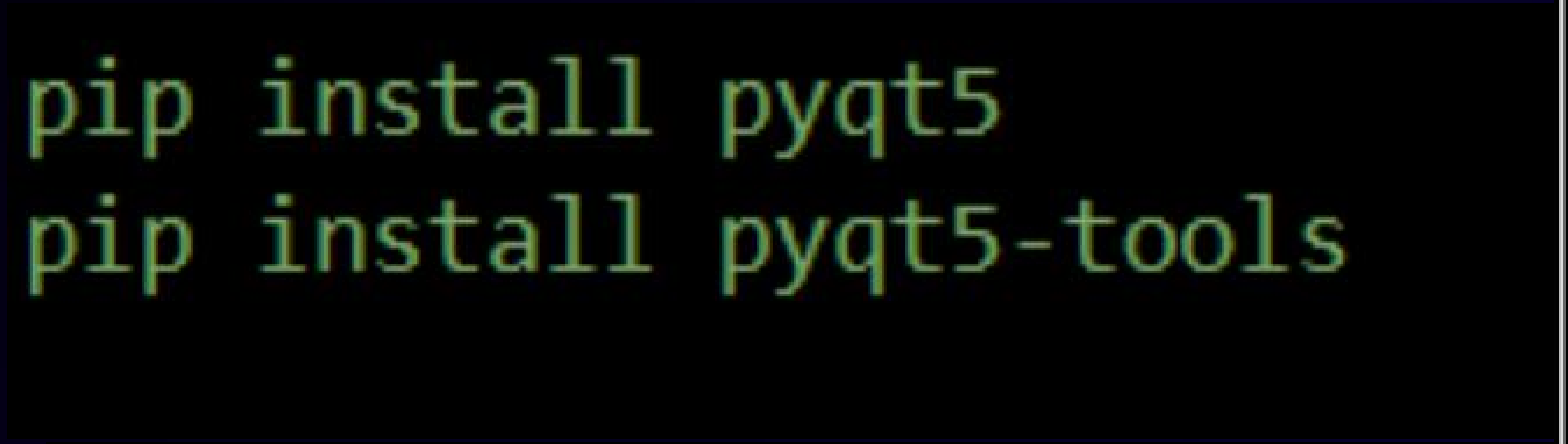
Rich Widgets Library: Offers various widgets like buttons, tables, sliders, and more.

ignal and Slot Mechanism

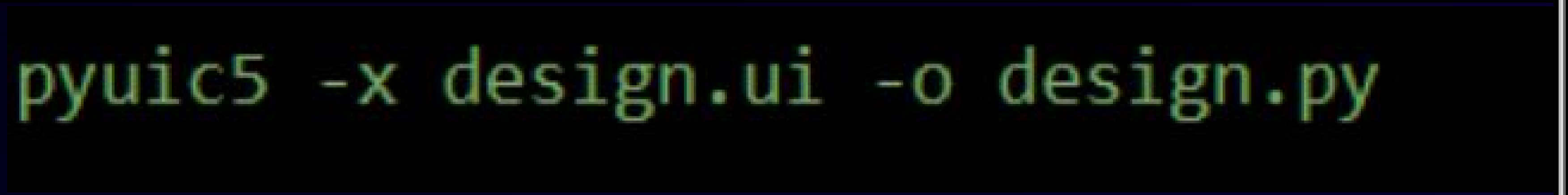
Integration with Qt Designer

Conecting pyqt with python,

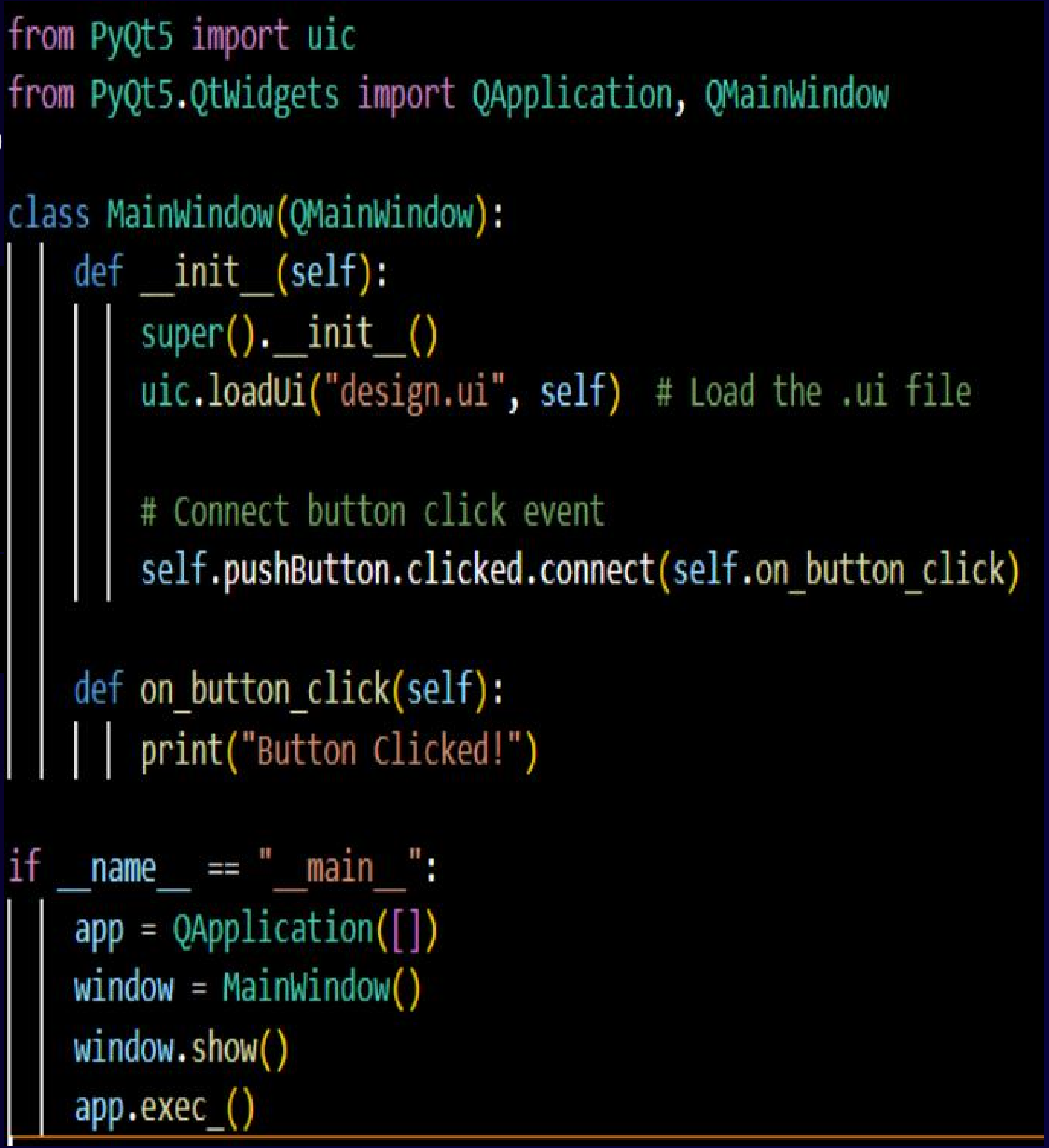
Convert a .ui file into a python scrip using the pyuic5 tool



```
pip install pyqt5
pip install pyqt5-tools
```



```
pyuic5 -x design.ui -o design.py
```



```python
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication, QMainWindow

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        uic.loadUi("design.ui", self)  # Load the .ui file

        # Connect button click event
        self.pushButton.clicked.connect(self.on_button_click)

    def on_button_click(self):
        print("Button Clicked!")

if __name__ == "__main__":
    app = QApplication([])
    window = MainWindow()
    window.show()
    app.exec_()
```

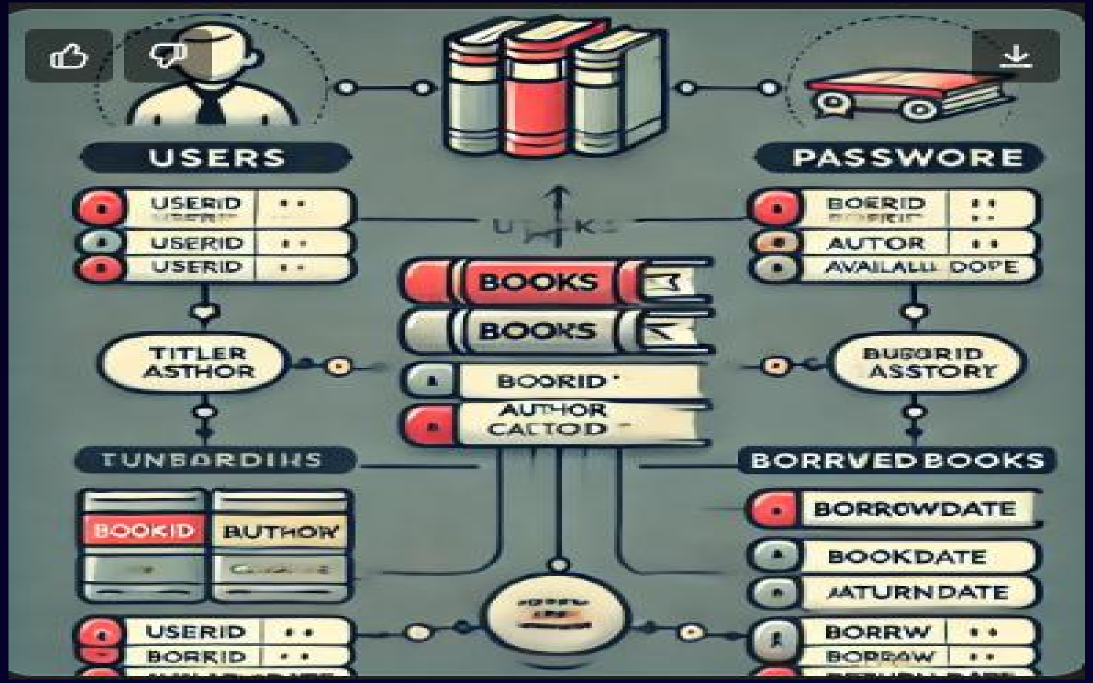
Purpose: Simplify library operations by digitizing user and book management.

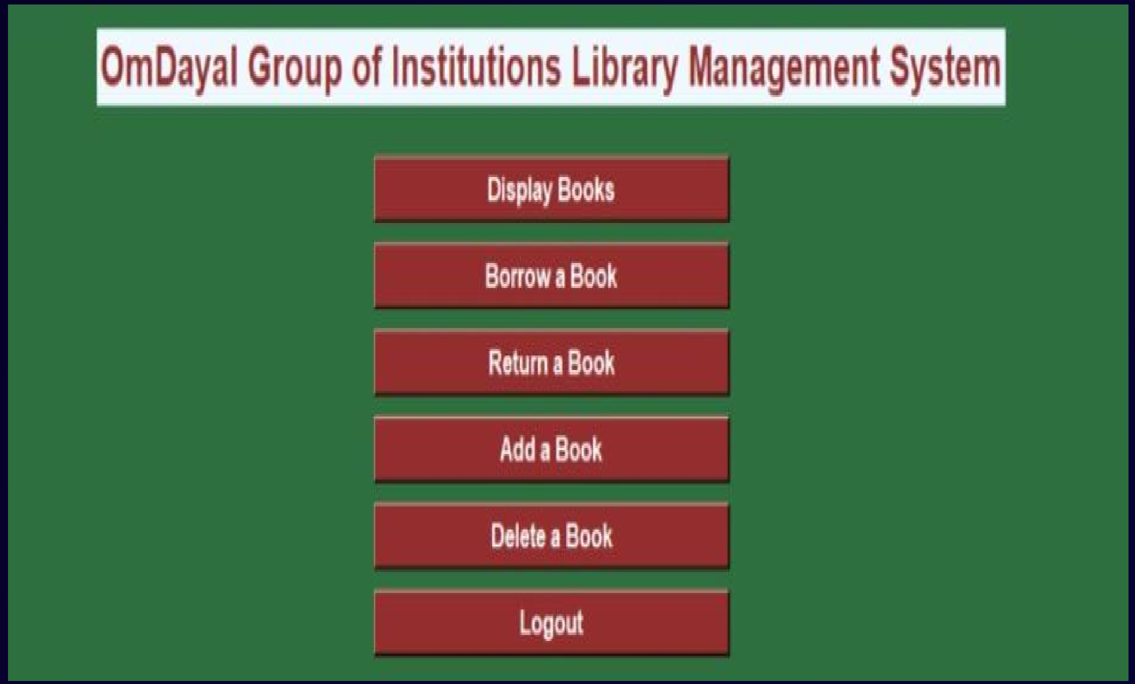
Key Highlights:

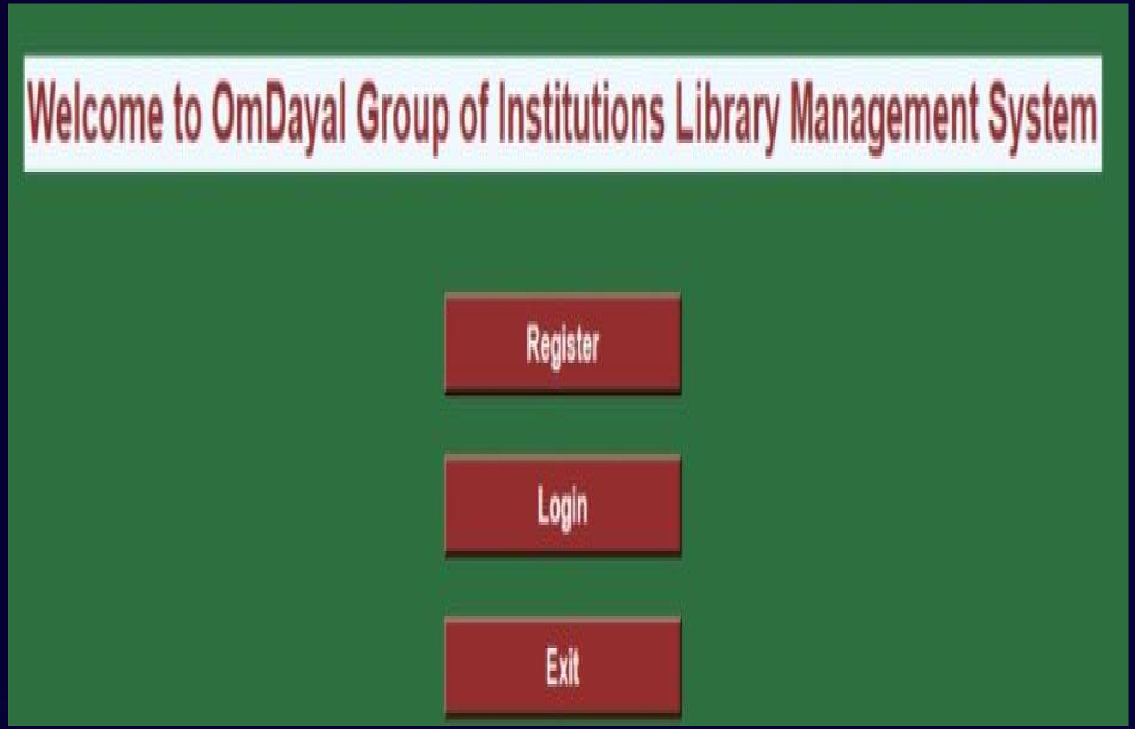
- Secure user login and registration.
- Real-time book borrowing and returning.
- Easy book management with category and publisher filters.

Features at a Glance

1. **User Authentication:** Secure login with password encryption.
2. **Book Management:** Add, delete, and view books by various attributes.
3. **Borrow/Return System:** Track transactions and manage availability.
4. **Data Security:** Ensures integrity with robust database design.






Welcome to OmDayal Group of Institutions Library Management System
Register
Login
Exit

•**User Registration & Login**: Secure access with username and password.

•**Book Management**: Add, delete, and display available books.

•**Book Borrowing & Returning**: Real-time tracking of borrowed and returned books.

•**Availability Check**: Ensures no borrowing of unavailable books.

UI Design:

•**Simple and User-Friendly Interface**: Clean layout with clear navigation and color scheme for easy access to features.

You can see our library manjmenst sytem project code

Gihhub link: https://github.com/SoumyodipThanadar/omdayallibrary.

Thank you!